#7
T.D.
12/11/03

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION FOR LETTERS PATENT

MULTIPLE SITE AUTOMATED LOGOUT

INVENTORS:

RYAN W. BATTLE

CHRISTOPHER E. MITCHELL

ATTORNEY'S DOCKET NO. MS1-826US

FIELD

[0001] This invention relates generally to the field of computers, and in particular to automatically logging out of multiple sites on computers.

COPYRIGHT NOTICE/PERMISSION

[0002] A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever. The following notice applies to the software and data as described below and in the drawing hereto: Copyright © 2000, Microsoft Corporation, All Rights Reserved.

BACKGROUND

[0003] The recent growth in popularity of the Internet has significantly increased the number of Internet users and the number of Internet sites (also referred to as "web sites"). Web sites may provide various types of information to users, offer products or services for sale, and provide games and other forms of entertainment. Many web sites require users to "register" by providing information about themselves before the web server grants access to the site. This registration information may include the user's name, account number, address, telephone number, email address, computer platform, age, gender, or hobbies. The registration information collected by the web site may be necessary to complete transactions (such as commercial or financial transactions). Additionally, information can be collected which allows the web site operator to learn about the visitors to the site to better target its future marketing activities or adjust the information provided on the web site. The collected information may also be used to

allow the web site to contact the user directly (e.g., via email) in the future to announce, for example, special promotions, new products, or new features of the web site.

[0004] When registering with a web site for the first time, the web site typically requests that the user select a login ID and an associated password. The login ID allows the web site to identify the user and retrieve the user's information during subsequent user visits to the web site. Generally, the login ID must be unique to the web site such that no two users have the same login ID. The password associated with the login ID allows the web site to authenticate the user during subsequent visits to the web site. The password also prevents others (who do not know the password) from accessing the web site using the user's login ID. This password protection is particularly important if the web site stores private or confidential information about the user, such as financial information or medial records.

[0005] If a user visits several different web sites, each web site may require entry of similar registration information about the user, such as the user's name, mailing address, and email address. This repeated entry of identical data is tedious when visiting multiple web sites in a short period of time. Many web sites require the user to register before accessing any information provided on the web site. Thus, the user must enter the requested registration information before they can determine whether the site contains any information of interest.

[0006] After registering with multiple web sites, the user must remember the specific login ID and password used with each web site or other Internet service. Without the correct login ID and password, the user must re-enter the registration information. A particular user is likely to have different login IDs and associated passwords on different web sites. For example, a user named Bob Smith may select "smith" as his login ID for a particular site. If the site already has a user with a login ID of "smith" or requires a login ID of at least six characters, then the user must select a different login ID.

[0007] After registering at numerous web sites, Bob Smith may have a collection of different login IDs, such as: smith, smith1, bsmith, smithb, bobsmith, bob_smith, and smithbob. Further, different passwords may be associated with different login IDs due to differing password requirements of the different web sites (e.g., password length requirements or a requirement that each password include at least one numeric character). Thus, Bob Smith must maintain a list of web sites, login IDs, and associated passwords for all sites that he visits regularly.

[0008] A few services provide a login type of service to make signing into accounts at multiple sites more convenient. Some services provide a key ring, which is essentially a set of images or icons (keys) which when selected provide ID and passwords to a site associated with the key. Each key may correspond to a different site. Another service provides a link to each desired site, which when selected logs a user into the site, and also contains further background information which may be required by a site. While such services make signing into sites easy, it is more difficult to easily sign out of the sites without visiting each site. In addition, the information required by the sites may be incorporated into data files referred to as cookies, which may or may not be deleted on logging out.

[0009] There is a need to be able to log out of multiple accounts, and not have other users of the same computer to get into the same accounts. Furthermore, with kiosks and other public devices that are used to access the web to check email, stocks and order from retail outlets, privacy is a great concern. There also is a need for some form of assurance that the user is logged out of each site. A user can manually delete sensitive information, but there is a danger of deleting personalization files. If such files are deleted, they must be entered again the next time the account is entered.

SUMMARY

[0010] A logout feature of a service that facilitates login to multiple domain websites maintains a list of the sites that a user logs on to during a session and completely logs the user out of all the sites they visited during the session.

[0011] A cookie named "Visited Sites" is used by a login server to maintain a list of all sites that a user logs on to during a session. When the user selects a logout link anywhere on the network, they are directed to a logout page on the login server. The login server retires all login domain cookies first, and displays a page that explains to the user that they are about to be logged out of each domain. The logout page generates image tags for each of the sites listed in the visited-sites cookie. The image tag provides a URL hosted at each site that expires cookies that are present in the user's cookie cache by setting their value to nothing, and their expiration date to a past date.

[0012] When a request to logout is received, the Visited Sites cookie is checked, and if present, it is read. All local cookies are then expired, and indicated as such on a user interface. In order to avoid redirecting the user to each domain, the logout page provides an individual image source for each site the user signed into. This enables the login server to log the user out of each domain by clearing selected cookies stored by the domains. It also clears the cookies from the user's browser and then indicates on the logout page if the logout was successful for each domain. Finally, the domain from which the user selected to logout specifies the URL to which the user is redirected.

BRIEF DESCRIPTION OF THE DRAWINGS

[0013] Fig. 1 is a block diagram showing pertinent components of a computer in accordance with the invention.

[0014] Fig. 2 illustrates an exemplary network environment in which the present invention is utilized.

[0015] Fig. 3 is a block diagram showing a browser, logout server, and two affiliate sites to which a user is logged into.

[0016] Fig. 4 is a flowchart indicating the logical flow of the logout server.

DETAILED DESCRIPTION

[0017] In the following detailed description of exemplary embodiments of the invention, reference is made to the accompanying drawings which form a part hereof, and in which is shown by way of illustration specific exemplary embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be utilized and that logical, mechanical, electrical and other changes may be made without departing from the spirit or scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

[0018] The detailed description is divided into multiple sections. A first section describes a simple representation of a computer system and the operation of multiple computer systems on a network which implement different aspect of the current invention. This is followed by a description of the invention and how it is implemented.

HARDWARE AND OPERATING ENVIRONMENT

[0019] An exemplary system for implementing the invention includes a computing device, such as computing device 100 in Figure 1. In its most basic configuration, computing device 100 typically includes at least one processing unit 102 and memory 104. Depending on

the exact configuration and type of computing device, memory 104 may be volatile (such as RAM), non-volatile (such as ROM, flash memory, etc.) or some combination of the two. This most basic configuration is illustrated in Figure 1 by broken line 106.

[0020] Device 100 may also include additional features/functionality. For example, device 100 may include additional storage (removable and/or non-removable) including, but not limited to, magnetic or optical disks or tape. Such additional storage is illustrated in Figure 1 by removable storage 108 and non-removable storage 110. Computer storage media includes volatile and nonvolatile, removable and non-removable media implemented in any method of technology for storage of information such as computer readable instructions, data structures, program modules or other data. Memory 104, removable storage 108 and non-removable storage 110 are all examples of computer storage media. Computer storage media includes, but is not limited to RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical storage, magnetic based storage or any other medium which can be used to store desired information and which can be accessed by device 100. Any such computer storage media may be part of device 100.

[0021] Device 100 may also contain communications connection(s) 112 that allow the device to communicate with other devices. Communications connection(s) 112 is an example of communication media. Communications media typically embodies computer readable instructions, data structures, program modules or other data in a modulated data signal such as a carrier wave or other transport mechanism and includes any information delivery media. The term "modulated data signal" means a signal that has one or more of its characteristics set of changed in such a manner as to encode information in the signal. By way of example, and not limitation, communication media includes wired media such as wired network or direct-wired connection, and wireless media such as acoustic, RF, infrared and other wireless media. The

term computer readable media as used herein includes both storage media and communications media.

[0022] Device 100 may also have input device(s) 114 such as keyboard, mouse, pen, voice input device, touch input device, etc. Output device(s) 116 such as display, speakers, printers, etc may also be included. All these devices are well known in the art.

[0023] This invention may be described in the context of computer-executable instructions, such as program modules, executed by one or more computer or other devices such as device 110. Generally, program modules include routines, programs, objects, components, data structures, etc. that perform particular tasks or implement particular abstract data types. Typically the functionality of the program modules may be combined or distributed as desired in various embodiments.

[0024] Fig. 2 is a block diagram illustrating an exemplary network environment in which the present invention is utilized. A client computer system 200 is coupled to a network 202. In this example, network 202 is the Internet (or the World-Wide Web). However, the teachings of the present invention can be applied to any data communication network that implements a stateless protocol similar to hypertext transfer protocol, http. Multiple affiliate servers 204, 206, and 208 are coupled to network 202, thereby allowing client computer system 200 to access web servers 204, 206, and 208 via the network. Affiliate servers 204, 206, and 208 are also referred to as "web servers", "network servers" and "sites" hosting content such as text and images for access by other computers on the network 202. An authentication server 210 is also coupled to network 202, facilitating communication between the authentication server and client computer system 200 and authentication servers 204, 206, and 208. Although referred to as an "authentication server", authentication server 210 is also a web server capable of interacting with web browsers and other web servers. In this example, data is communicated

between the authentication server 210, client computer system 200, and web servers using http, a protocol commonly used on the Internet to exchange information. An http specification is published by the Internet Engineering Task Force.

[0025] An authentication database 212 is coupled to authentication server 210. The authentication database 212 contains information necessary to authenticate users and also identifies which elements of user profile information should be provided to a particular affiliate server when the user accesses the affiliate server. Although the authentication database 212 is shown separately from the authentication server 210, in other embodiments of the invention, the authentication database is contained within the authentication server.

[0026] An authentication process authenticates a user of client computer 200 seeking access to an affiliate server 204, 206, or 208. The authentication server 210 authenticates the user of client computer 200 by requesting authenticating information, such as the user's login ID and password. If the user is successfully authenticated, then authentication server 210 generates an encrypted authentication ticket and communicates the ticket to the appropriate affiliate server. The authentication ticket indicates that the user is authenticated. Each affiliate server requires a key in order to decrypt the ticket and allow access by the user.

[0027] The authentication ticket contains two time stamps. The first time stamp indicates the last time that the user's login ID and password were physically typed by the user. The second time stamp indicates the last time that the user's login information was refreshed by the authentication server. This "refresh" of the user's login information can be performed "silently" or by manual entry of the login information (i.e., login ID and password) by the user. The refreshing of the user's login information is performed by the authentication server. Once completed, a new authentication ticket is issued to the affiliate server indicating the new time stamp values.

[0028] The term "affiliate server" is defined herein as a web server that has "registered" or established a relationship or affiliation with the authentication server 210. Each affiliate server 204, 206, and 208 includes a code sequence that allows the affiliate server to communicate with the authentication server 210 when a user (who is also registered with the authentication server) requests access to the affiliate server.

[0029] Prior to executing the authentication process, both the user of client computer system 200 and the operator of affiliate server 204 "register" with the authentication server 210. This registration is a one-time process which provides necessary information to the authentication server. The user of client computer system 200 registers by providing information such as the user's email address, password information, and various other information about the user or the client computer system if desired. As part of the user registration process, the user is assigned (or selects) a login ID, which is a common login ID used to access any affiliate server. The login ID may also be referred to herein as a "user name" or "login name". Additionally, the user selects a password associated with the login ID which is used for authentication purposes.

[0030] After registering and logging into the authentication server, the user can visit any affiliate server (i.e., affiliate servers that are also registered with the same authentication server) without requiring any additional authentication and without re-entering user information that is already contained in the associated user profile.

[0031] The operator of affiliate server 204 registers with the authentication server 210 by providing information about the affiliate server (e.g., server name and internet address). Additionally, the affiliate server provides information regarding its authentication requirements. The authentication requirements can be specified as the maximum time allowed since the last login and entry of authentication information by the user as well as the maximum time allowed since the last "refresh" of the authentication information by the user. Refreshing the

authentication information refers to the process of having the user re-enter the password to be certain that the appropriate user is still operating the client computer system. This periodic refreshing of authentication information is useful if the user leaves their computer system without logging out of the authentication server, thereby allowing another individual to access affiliate servers using the login ID of the previous user. If a user requests access to the affiliate server after the maximum time allowed, then the user is re-authenticated (i.e., refreshed) by the authentication server by issuing a new authentication ticket either silently or with required reentry of password as described above. Thus, although there is a central authentication server, each individual affiliate server can establish its own authentication requirements which are enforced by the authentication server. After registering with the authentication server, the affiliate server can use the authentication server to authenticate any user that has also registered with the authentication server.

[0032] When first logging into an affiliated server web page, the user is redirected to the authentication server for entry of ID and password. If the user-entered information is correct (i.e., matches the information stored in the authentication database) then the authentication server sets appropriate cookies in the client computer system and redirects the user's browser to the affiliate server. A "cookie" is a piece of data provided to a web browser by a web server. The data (i.e., cookie) is sent back to the web server by the web browser during subsequent accesses to the web server. One cookie contains information regarding the date and time that the user was authenticated by the authentication server. Another cookie contains information regarding the user profile. The authentication server also updates (or creates) a cookie that contains a list of all sites (or web servers) visited by the user since the last logout from the authentication server. This cookie is referred to as a visited sites cookie. The visited sites cookie is updated by adding

the current affiliate server to the list of sites visited. The list may consist of a set of site IDs which were assigned at registration.

[0033] Due to security features implemented in current web browsers, and in compliance with the http specification, cookies written to the client computer system by the authentication server cannot be read by any affiliate server. Similarly, cookies written to the client computer system by a particular affiliate server cannot be read by any other affiliate server. The cookies written by an affiliate server are encrypted using a key that is unique to the affiliate server, thereby preventing other affiliate servers from reading the data stored in the cookies.

[0034] The authentication server also communicates the user profile information to the affiliate server through the client computer system. In a particular embodiment of the invention, the user of the client computer system can specify during the registration process what types of profile information should be provided to various types of web servers. For example, a user may specify that all commerce-related web servers should receive the user's email mail address, but restrict the mailing address from all other types of web sites.

[0035] After receiving the authentication ticket and the user's profile information, the affiliate server may generate a personalized web page for the user and communicates the web page to the user's browser. Additionally, the affiliate server copies one or more cookies to the client computer system which include information indicating that the user of the client computer system has been authenticated and indicating the period of time during which the authentication is valid. Each time the user enters a new web page request on the same affiliate server, the data in the cookie is sent to the affiliate server along with the page request. Thus, the affiliate server will not repeatedly check the authentication of a user during each subsequent page request. However, if a particular period of time has passed (referred to as a timeout period) since the last

authentication process by the authentication server, then the affiliate server may request a re-authorization of the user.

[0036] It is difficult for a user to individually remove each cookie when logging out of an affiliate server, or when logging out of all affiliate servers in the visited sites cookie. Since one server cannot access the cookies provided by another server, each affiliate server individually logs out each user.

[0037] In Figure 3, an improved way of logging a user out of each affiliate server indicated in the visited sites cookie is shown in block diagram. A browser is shown at 310 in communication with a login server 320. The browser 310 is also shown as communicating with two affiliated servers 330 and 340. The affiliated servers for purposes of this description each reside on a different domain. It should be recognized that a domain may have more than one server, and that logging into one server may log a user into multiple servers on the same domain.

[0038] When a user desires to log out, the user can go to any site that contains a logout link pointing to a logout page on the login server 320. This results in the browser issuing a get visited sites cookie to the login server 320. The logout service on the authentication server builds an image source tag corresponding to each site identified in the visited sites cookie by providing markup language based source to the browser as indicted at 350. Each image source tag has a site ID. Three site IDs are shown at 350, "10, 15 and 7000". The visited sites cookie may simply be the list of site IDs in text string such as: "10, 15, 7000".

[0039] The image source tags are rendered into a page on the browser 310 that identifies the sites in the visited sites cookie, and also provides a position for a check mark or "x" mark to indicate whether or not logout was successful for each visited site. Text is provided at the top of the page in this example: "Please wait while we sign you out". A more elaborate page may easily be created if desired. Further, text may be provided instructing the user to wait for all

check marks to render, and try again later or go to individual sites directly to ensure proper logout.

[0040] Each image source tag has a query string parameter on the end of it to cause the browser go and fetch the image via a separate transaction as opposed to referring to a cache of the image. The image source tag points to an expire cookies uniform resource locator (URL) that is hosted by each affiliated server. It responds with a set cookie header that clears desired cookies from the browser. One manner of clearing such cookies is to set their value to nothing, and set their expire time to a past date. The particular cookies cleared include profiles and authorizations, as well as site cookies that were generated by the visited affiliated servers. The cookies are labeled MSPAuth (authorization), MSPProf (profile), and site cookies in Figure 3. Affiliated servers also respond with a small checkmark image which is added to the logout page on the browser to indicate successful logout. In one embodiment, code is used to look for a response from each affiliated server, and if not received in a desired period of time, it places an "x" or other symbol indicative of logout failure. Finally, the affiliated server from which logout was selected may be allowed to specify the page to be displayed to the user.

[0041] One aspect of the invention involves tricking the browser to think it is fetching image source from the domains. The browser is simply issuing separate requests for images to each visited affiliated server. The affiliated servers believe that they are returning just an image, but also send the set cookie header along with the checkmark which causes the browser to delete the desired cookies.

[0042] Figure 4 is a flowchart of the logical flow of the logout process. At 410, the user selects logout on any affiliate server, or the authentication server. A non-secure connection is started at 415, and an incoming request to logout is received by the login server at 420. The request indicates the site ID which was assigned during registration, and may contain many of

the same arguments used to log in, such as time windows and current time as well as other information if desired. It also may identify a return URL to indicate a page to direct the browser to when done logging out. At 425, a check is made to determine if the visited sites cookie is present. If not, an error case occurs and is handled at 430. If a site ID is provided, a logout user interface is rendered with a site ID expire cookie URL image tag that was specified by the site during registration. If no site ID is provided, a logout user interface indicating that "You are Signed Out" is provided. The user browser is then redirected to the domain on which logout was selected.

[0043] If at 425, a visited sites cookie is found, it is read at 435. At 440, a check is made to determine if the site ID has specified a logout URL during registration. If not, all local cookies are expired and a logout user interface is rendered with all siteIDs expire cookie URL image tags with a timeout upon which the browser is redirected to the domain's return URL.

[0044] If at 440, a siteID has a logout URL, all local cookies are expired and a logout user interface is rendered with all siteIDs expire cookie URL image tags with a timeout upon which the browser is redirected to the siteID logout URL.

[0045] The invention provides a simple mechanism to facilitate a user signing out of multiple sites without requiring special client downloads, server to server communication or special user interaction. The invention makes it easier to securely visit password protected sites while traveling and using kiosks, or generally using any browser that multiple people may access. It may be used simply for convenience, but also may be used to minimize the risk of compromising the users passwords or inadvertently providing personal or sensitive information to others.